

Managing Certificates in your Nexus Dashboard, Release 4.1.1

Table of Contents

New and changed information . . . 1
Understanding certificate management . . . 2
  LAN deployments . . . 2
  SAN deployments . . . 2
Handling pass phrases . . . 3
Generate keys . . . 4
  Generate a private key, create a CSR, and obtain a CA-signed certificate . . . 4
    Understanding RSA and ECDSA private keys . . . 4
    Generating a private key, create a CSR, and obtain a CA-signed certificate . . . 4
  Generate a private key and a self-signed certificate . . . 7
CA certificates . . . 11
  About NX-API and bootstrap certificates . . . 11
  Certificate generation and management . . . 11
    Certificate validation . . . 12
    Troubleshoot certificate validation failures . . . 12
    NX-API certificate verification by Nexus Dashboard . . . 14
  Upload CA certificates . . . 15
  Install certificate bundles on a bootstrap bench router . . . 15
  Delete CA certificates . . . 15
  Switch NX-API certificates . . . 16
    Upload certificates . . . 16
    Assign switches and install certificates . . . 16
    Unlink and delete certificates . . . 16
  Bootstrap certificates . . . 17
System certificates . . . 18
  Upload system certificates . . . 18
  Map a feature to a system certificate . . . 18
  Delete system certificates . . . 19
Fabric certificates . . . 20
  Enable NX-API certificate verification . . . 20
  Fabric certificates . . . 20
    Guidelines and limitations: Fabric certificates . . . 20
    Upload fabric certificates . . . 20
    Delete fabric certificates . . . 21

New and changed information

The following table provides an overview of the significant changes up to this current release. The table does not provide an exhaustive list of all changes or of the new features up to this release.

Release Version	Feature	Description
Nexus Dashboard 4.1.1	Improved navigation and workflow for managing certificates in your Nexus Dashboard	Beginning with Nexus Dashboard 4.1.1, the navigation and workflow for managing certificates in your Nexus Dashboard have been enhanced.

Understanding certificate management

Certificate management differs based on the type of deployment in your Nexus Dashboard and the user role.

LAN deployments

- For users with the super admin role, all certificate management options are enabled.
- For users with the fabric admin role and “All” security domain, all of tasks are enabled except **System Certificates**. For system certificates, the list API is allowed.
- For any remaining user roles, only the certificates list is available.

SAN deployments

- For users with the super admin role, **CA Certificates** and **System Certificates** options are available with the following criteria:
  - In the **CA Certificate** page, **Install** on a device is disabled.
  - In the **System Certificate** page, **Manage feature attachments**, “bootStrapServer”, “messageBus” and “cyberArk” is disabled.
  - The **Fabric Certificates** page is not available.
- For users with the fabric admin role and “All” security domain, the behavior is the same as that for users with the super admin role.
- For any remaining user roles, only the certificates list is available.

Handling pass phrases

At certain points in the certificate management process, such as when you are generating a key, the process prompts you to provide a mandatory PEM pass phrase.

When you enter a pass phrase as part of the certificate management process, do not lose the pass phrase information. If you lose the pass phrase, you will not be able to upload the certificates later in the process. Cisco is not able to aid you if you lose the pass phrase related to the certificate management process.

Generate keys

Generate a private key, create a CSR, and obtain a CA-signed certificate

This section provides an example of how to generate a private key, create a certificate signing request (CSR), and obtain a certificate signed by a Certificate Authority (CA) for use in your Nexus Dashboard cluster. If you want to generate both a key and a self-signed certificate, skip this section and follow the steps described in Generate a private key and a self-signed certificate instead. The configuration steps required to add the keys and certificates in the Nexus Dashboard GUI are described later in this article.

Understanding RSA and ECDSA private keys

In previous releases, only RSA private keys were supported. Beginning with Nexus Dashboard release 4.1.1, both RSA and ECDSA private keys are supported.

The most widely used curves are secp256r1 (also known as P-256) and secp256k1, with the latter being popular in cryptocurrency applications.

Following are the recommended key lengths (based on recent NIST guidelines):

- For general-purpose security: Use 2048-bit RSA.
- For long-term or high-security needs: Use 3072-bit RSA or higher.
- For highest security requirements: Consider 4096-bit RSA, or switch to ECC (Elliptic Curve Cryptography), which provides equivalent security at much shorter key lengths (such as secp256r1 ECC ≈ 3072-bit RSA).

Generating a private key, create a CSR, and obtain a CA-signed certificate

1. Generate a private key and create the CSR configuration file (csr.cfg) with the required information.

   See Understanding RSA and ECDSA private keys for more information.

   - To generate a private key and create a CSR using an **RSA** private key:

     a. Generate a 2048-bit RSA private key and encrypt it using AES-256:

     ```
     openssl genpkey -algorithm RSA -aes256 -out nexus-dashboard-private.key
     -pkeyopt rsa_keygen_bits:2048
     ```

     

     You will be prompted to enter a mandatory PEM pass phrase after you enter this command. See Handling pass phrases for more information on handling pass phrases.

     b. Create a CSR using the AES-encrypted private key and sign it with SHA-256:

```
openssl req -new -key nexus-dashboard-private.key -out nexus-dashboard-private.csr -sha256
```

c. Create the CSR configuration file (csr.cfg) with the required information.

An example configuration file is shown below:

```
[req]
default_bits = 2048
distinguished_name = req_distinguished_name
req_extensions = req_ext
prompt = no
[req_distinguished_name]
countryName = US
stateOrProvinceName = Texas
localityName = Plano
organizationName = CSS
organizationalUnitName = DC
commonName = nd.dc.css
emailAddress = no-reply@mydomain.com
[req_ext]
subjectAltName = @alt_names
[alt_names]
DNS.1 = athos.dc.edu
IP.1 = 10.0.0.96
DNS.2 = porthos.dc.edu
IP.2 = 10.0.0.97
DNS.3 = aramis.dc.edu
IP.3 = 10.0.0.98
```

d. Specify your parameter file when generating the CSR:

```
openssl req -newkey ec:<(openssl genpkey -genparam -algorithm ec -pkeyopt ec_paramgen_curve:P-256) -keyout PRIVATEKEY.key -out MYCSR.csr -config csr.cfg
```

- To create an **ECDSA** private key with your CSR:

a. Generate a parameter file for a 256-bit ECDSA key:

```
openssl genpkey -genparam -algorithm ec -pkeyopt ec_paramgen_curve:P-256 -out ECPARAM.pem
```



You will be prompted to enter a mandatory PEM pass phrase after you enter this command. See Handling pass phrases for more information on handling pass phrases.

b. Specify your parameter file when generating the CSR:

```
openssl req -newkey ec:<(openssl genpkey -genparam -algorithm ec -pkeyopt
ec_paramgen_curve:P-256) -keyout PRIVATEKEY.key -out MYCSR.csr
```

3. Obtain a CA-signed certificate.

   In production deployments, you will provide the CSR (nd.csr) from the previous step to a public CA, such as IdenTrust or DigiCert, to obtain the CA-signed certificate (nd.crt).

4. Verify the signed certificate.

   The following command assumes you copied the CA-signed certificate (ca.crt) into the same folder as the private key you generated.

```
[rescue-user@localhost ~]$ openssl verify -verbose -CAfile ca.crt nd.crt
nd.crt: OK
```

5. Rename the signed certicate with the name on the private key.

   The server certificate signed through CA and the private key file must share the same name.

6. After obtaining the signed certificate, configure a certificate chain.

   To use multiple certificates, place any intermediate certificates after the server certificate and before the root certificate. You can add as many intermediate certificates as you need, in decreasing order of hierarchy, up to the root certificate.

7. Add the contents of the generated files in your Nexus Dashboard's GUI.
   a. Navigate to **Admin > Certificate Management**.
   b. Click **CA Certificates**.
   c. Click **Add CA certificate** to upload the chained certificate.
8. Upload these files to **Admin > Certificate Management > System Certicates** and enter the necessary password.
9. Attach and install the certificate.
   - In the row **Manage Feature Attachments**, click the ellipse (...) and choose **Webserver** to attach and install the certificate,

     or

   - Choose the certificate to see the **Manage Feature Attachments** button, then click to choose **WebSever**, then click **Save**.

Generate a private key and a self-signed certificate

This section provides an example of how to generate a private key and custom certificates should you want to use them in your Nexus Dashboard cluster.

If you want to use a CA-signed certificate, skip this section and follow the steps described in Generate a private key, create a CSR, and obtain a CA-signed certificate.

The configuration steps required to add the keys and certificates in the Nexus Dashboard GUI are described in the section " Security Configuration" in the Configuring Users and Security.

1. Generate private key with PEM passphrase.

   You can generate the private key on any platform that has OpenSSL installed or you can SSH into one of your Nexus Dashboard nodes as the rescue-user and perform these steps there.

   ```
   [rescue-user@localhost ~]$ openssl genpkey -algorithm RSA -aes256 -out nd.key
   -pkeyopt rsa_keygen_bits:2048
   ..........................
   ..............
   e is 65537 (0x10001)
   [rescue-user@localhost ~]$ ls
   nd.key
   ```

   

   You will be prompted to enter a mandatory PEM passphrase after you enter this command. See Handling pass phrases for more information on handling pass phrases.

2. Generate Certificate Authority (CA) key.

   To generate a self-signed CA, for example for lab and testing purposes, run the following command:

   ```
   [rescue-user@localhost ~]$ openssl genrsa -out ca.key 2048
   Generating RSA private key, 2048 bit long modulus
   ...................
   ..............
   e is 65537 (0x10001)
   [rescue-user@localhost ~]$ ls
   ca.key  nd.key
   ```

3. Generate CSR for the CA.

   ```
   [rescue-user@localhost ~]$ openssl req -new -key ca.key -subj
   " /CN=Self/C=US/O=Private/ST=Texas"  -out ca.csr
   [rescue-user@localhost ~]$ ls
   ```

```
ca.csr  ca.key  nd.key
```

You can view the generated CSR using the following command:

```
[rescue-user@localhost ~]$ openssl req -in ca.csr -text -noout
```

4. Create self-signed root certificate.

```
[rescue-user@localhost ~]$ openssl x509 -req -in ca.csr -signkey ca.key
-CAcreateserial -out ca.crt -days 3650
Signature ok
subject=/CN=Self/C=US/O=Private/ST=Texas
Getting Private key
[rescue-user@localhost ~]$ ls
ca.crt  ca.csr  ca.key  nd.key
```

You can view the generated root certificate using the following command:

```
[rescue-user@localhost ~]$ openssl x509 -in ca.crt -text -noout
```

5. Generate your CSR signed with the private key you generated in the first step.
   a. Create the CSR configuration file (csr.cfg) with the required information.

   An example configuration file is shown below:

```
[req]
default_bits = 2048
distinguished_name = req_distinguished_name
req_extensions = req_ext
prompt = no
[req_distinguished_name]
countryName = US
stateOrProvinceName = Texas
localityName = Plano
organizationName = CSS
organizationalUnitName = DC
commonName = nd.dc.css
emailAddress = no-reply@mydomain.com
[req_ext]
subjectAltName = @alt_names
[alt_names]
DNS.1 = *.dc.css
IP.1 = 10.0.0.96
```

```
IP.2 = 10.0.0.97
```

b. Use the new encrypted private key from Step 1.

```
[rescue-user@localhost ~]$ openssl req -new -key nd.key -out nd.csr -config csr.cfg
-sha256
[rescue-user@localhost ~]$ ls
ca.crt  ca.csr  ca.key  csr.cfg  nd.csr  nd.key
```

You can view the generated CSR using the following command:

```
[rescue-user@localhost ~]$ openssl req -in nd.csr -text -noout
```

6. Self-sign the certificate you generated.

```
[rescue-user@localhost ~]$ openssl x509 -req -in nd.csr -CA ca.crt -CAkey ca.key
-CAcreateserial -out nd.crt -days 3600
Signature ok
subject=/C=US/ST=Texas/L=Plano/O=CSS/OU=DC/CN=nd.dc.css/emailAddress=no-
reply@mydomain.com
Getting CA Private Key
[rescue-user@localhost ~]$ ls
ca.crt  ca.csr  ca.key  ca.srl  csr.cfg  nd.crt  nd.csr  nd.key
```

7. Verify the signed certificate.

```
[rescue-user@localhost ~]$ openssl verify -verbose -CAfile ca.crt nd.crt
nd.crt: OK
```

8. After obtaining the signed certificate, configure a certificate chain.

   To use multiple certificates, place any intermediate certificates after the server certificate and before the root certificate. You can add as many intermediate certificates as you need, in decreasing order of hierarchy, up to the root certificate.

9. Add the contents of the generated files in your Nexus Dashboard's GUI.
   a. Navigate to **Admin > Certificate Management**.
   b. Click **CA Certificates**.
   c. Click **Add CA certificate** to upload the chained certificate.
10. Rename the signed certicate with the name on the private key.

    The server certificate signed through CA and the private key file must share the same name.

11. Upload these files to the **Admin > Certificate Management > System Certicates** and enter the necessary password.
12. Attach and install the certificate.
    - In the row **Manage Feature Attachments**, click the ellipse (...) and choose **Webserver** to attach and install the certificate,

      or

    - Choose the certificate to see the **Manage Feature Attachments** button, then click to choose **WebSever**, then click **Save**.

CA certificates

About NX-API and bootstrap certificates

Cisco NX-OS switches require an SSL certificate to function in NX-API HTTPS mode. You can generate the SSL certificates and get it signed by your CA. You can install the certificates manually using CLI commands on switch console or use Nexus Dashboard to install these on switches.

Nexus Dashboard provides a Web UI framework to upload NX-API certificates to Nexus Dashboard. Later, you can install the certificates on the switches that are managed by Nexus Dashboard.



For Nexus switches, this feature is supported on switches running Cisco NXOS version 9.2(3) or higher.

Certificate generation and management

For each switch, the data center administrator generates an ASCII (base64) encoded certificate. This certificate comprises two files:

- .key file that contains the private key
- .crt/.cer/.pem file that contains the certificate

Nexus Dashboard also supports a single certificate file that contains an embedded key file, that is, the .crt/.cer/.pem file, which can also contain the contents of the .key file. Beginning with Nexus Dashboard release 4.1.1, you must provide separate certificates and key files when attaching them. The key files are accepted if encrypted with a password.

Note that .pem files are typically in PEM format and contain the certificate or key encoded in Base64, while .crt files are usually in PEM format (Base64 encoded) or DER format (binary encoded).

If any of your devices accepts only .pem format, and:

- You have a .crt file that is in *PEM* format, then enter the following to create a .pem file:

```
cp certificate.crt certificate.pem
```

- If you have a .crt file that is in *DER* format, then enter the following to convert it to PEM:

```
openssl x509 -inform DER -in certificate.crt -out certificate.pem
```

You can either use CA-signed certificates or self-signed certificates. Cisco Nexus Dashboard does not mandate the signing; however, the security guidelines suggest you use the CA-signed certificates.

You can generate multiple certificates for multiple switches, to upload to Nexus Dashboard. Ensure that you name the certificates appropriately, to help you choose the switch meant for that certificate.

You can upload one certificate and the corresponding key file, or bulk upload multiple certificates and

key files. After the upload is complete, you can view the upload list before installing these on the switches. If a certificate file that contains an embedded key file is uploaded, Nexus Dashboard derives the key automatically.

Certificate and the key file must have the same filename. For example, if a certificate filename is mycert.pem, the key filename must be mycert.key. If the certificate and key pair filenames are not the same, then Nexus Dashboard will not be able to install the certificate on the switch.

Nexus Dashboard allows you to bulk install the certificates to the switches. Because bulk installation uses the same password, all encrypted keys must be encrypted with the same password. If the password is different for a key, you cannot install the certificate in bulk mode. Bulk mode installation allows you to install encrypted and unencrypted keys certificates together, but all the encrypted keys must have the same password.

When you install a new certificate on the switch, it replaces the existing certificate with the new certificate.

You can install the same certificate on multiple switches; however, you cannot use the bulk upload feature.



Nexus Dashboard does not enforce the validity of certificates or options provided in it. It is up to you and the requirements on the switch to follow the convention. For example, if a certificate is generated for Switch-1 but it is installed on Switch-2, Nexus Dashboard doesn't enforce it; switches may choose to accept or reject a certificate based on the parameters in the certificate.

Certificate validation

Nexus Dashboard release 4.1.1 introduces enhanced certificate validation that includes strict X.509 checks, aligning with industry-standard cryptographic practices. This enhanced validation enforces all X.509 options and extensions (including extended key usage (EKU)) according to CA rules, ensuring robust security.

While Nexus Dashboard does not specifically mandate a particular EKU, it rigorously enforces any EKU values present in a certificate. If a certificate contains supported X.509 extensions or options with non-compliant values, Nexus Dashboard may reject it. This means certificates that functioned in earlier Nexus Dashboard versions (3.2.x and earlier versions) may now be deemed invalid if they do not meet these stricter criteria.

For example, if a certificate is intended for a web server, the X509v3 EKU field must include *TLS Web Server Authentication* or be entirely absent. If the certificate does not meet these conditions, Nexus Dashboard marks it as invalid.

Troubleshoot certificate validation failures

If certificate validation fails, Nexus Dashboard displays a generic error message, such as cannot find matching CA, and it may not always specify the exact issue. This message often indicates that the certificate does not comply with the strict X.509 checks introduced in Nexus Dashboard release 4.1.1.

To inspect the details of your certificate, including its X509v3 extensions, use the following command:

```
openssl x509 -noout -text -in <certfile>
```

Review the output, particularly the X509v3 extensions section. For example, look for X509v3 EKU field. If this field is present, ensure it contains appropriate values like *TLS Web Server Authentication* for server certificates, or ensure it is absent if the certificate does not require a specific EKU for its purpose.

Example output (illustrating *TLS Web Server Authentication* present)

```
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number:
            xxxxxxxxxxxxxxxxxxxxxxxx
        Signature Algorithm: sha256WithRSAEncryption
        Issuer: C=US, ST=California, L=San Jose, O=Example Corp, CN=Example Root CA
        Validity
            Not Before: Sep 26 21:48:09 2025 GMT
            Not After : Dec 30 21:48:09 2027 GMT
        Subject: C=US, ST=California, L=San Jose, O=Example Corp, CN=Example Root CA
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                Public-Key: (2048 bit)
                Modulus:
                    xxxxxxxxxxxxxxxx
                Exponent: 65537 (0x10001)
        X509v3 extensions:
            X509v3 Basic Constraints: critical
                CA:TRUE
            X509v3 Key Usage: critical
                Digital Signature, Certificate Sign, CRL Sign
            X509v3 Extended Key Usage:
                TLS Web Server Authentication, TLS Web Client Authentication  <-- This line is
critical
            X509v3 Subject Key Identifier:
                xxxxxxxxxxxxx
            X509v3 Authority Key Identifier:
                xxxxxxxxxxxxx
    Signature Algorithm: sha256WithRSAEncryption
    Signature Value:
        Xxxxxxxxxxxxxxxxxxx
```

Example output (illustrating absence of X509v3 EKU extension)

This certificate is also valid for server authentication because the X509v3 EKU extension is not

present, meaning there are no explicit restrictions on its extended use.

```
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number:
            xxxxxxxxxxxxx
        Signature Algorithm: sha256WithRSAEncryption
        Issuer: C=US, ST=California, L=San Jose, O=Example Corp, CN=Example Root CA
        Validity
            Not Before: Sep 28 17:21:03 2025 GMT
            Not After : Jan  1 17:21:03 2028 GMT
        Subject: C=US, ST=California, L=San Jose, O=Example Corp, CN=Example Root CA
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                Public-Key: (2048 bit)
                Modulus:
                    xxxxxxxxxxxxx              Exponent: 65537 (0x10001)
        X509v3 extensions:
            X509v3 Basic Constraints: critical
                CA:TRUE
            X509v3 Key Usage: critical
                Digital Signature, Certificate Sign, CRL Sign
            X509v3 Subject Key Identifier:
                xxxxxxxxxxxxx          X
            509v3 Authority Key Identifier:
                xxxxxxxxxxxxx
    Signature Algorithm: sha256WithRSAEncryption
    Signature Value:
        xxxxxxxxxxxxx
```

NX-API certificate verification by Nexus Dashboard

Nexus Dashboard supports a capability to verify NX-API certificates offered by switches. The NX-API requests done by Nexus Dashboard require SSL connection, and switches act like SSL server and offer server certificate as part of SSL negotiations. If provided a corresponding CA certificate, Nexus Dashboard can verify it.



By default, NX-API certificate verification is not enabled because it requires all switches in the data center to have the CA-signed certificates installed, and Nexus Dashboard is fed all the corresponding CA certificates.

Nexus Dashboard NX-API certificate management provides two functionalities named as Switch Certificates and CA Certificates to manage the same.

Upload CA certificates

To upload the CA certificates onto Nexus Dashboard, perform the following steps:

1. Navigate to the **Certificate Management** window:

   Admin > Certificate Management

2. Click the **CA Certificates** tab, then click **Add CA certificate** to upload the appropriate license file.

   For Secure POAP enabled switches, you must upload Root CA Certificate files. You can upload multiple files at a single instance.

3. Browse your local directory and choose the certificate-key pair to upload.

   You can upload certificates with the .pem/.cer/.crt/ file extensions.

   

   Root CA certificates are public certificates and do not contain keys. Switches require these Root CA bundles to verify Nexus Dashboard POAP/PnP server certificate which is signed by one of the Root CA in the bundle.

4. Click **Save** to upload the selected files to Nexus Dashboard.

   A successful upload message appears. The uploaded certificates are listed in the table.

Install certificate bundles on a bootstrap bench router

Nexus Dashboard does not assign the Root CA certificate bundle to the Bench Routers. Hence after installing new certificates, ensure that you install the new certificates on the Bench Router (BR).

To install certificate bundles on bootstrap bench router (BR):

1. Navigate to the **Certificate Management** window:

   Admin > Certificate Management

2. Click the **CA Certificates** tab.
3. Choose the appropriate certificate, then click **Actions > Install Certificate Bundle to POAP Bench Router (BR)**.

   The **Install Certificate Bundle to Bootstrap Bench Router (BR)** window appears.

4. Click **Assign**, and choose the relevant switches in the **Assign** window.

Delete CA certificates

You can delete CA certificates after uploading new certificates.

1. Unassign the bench router to delete the certificate.
2. Click **Actions > Delete** to delete the certificate from Cisco Nexus Dashboard.

Switch NX-API certificates

Upload certificates

To upload the certificates onto Nexus Dashboard, perform the following steps:

1. Click **Upload Certificate** to upload the appropriate certificate file.
2. Browse your local directory and choose the certificate key pair that you must upload to Nexus Dashboard.

   You can choose certificates with extension .cer/.crt/.pem + .key file separately.

3. Click **Upload** to upload the selected files to Nexus Dashboard.

   A successful upload message appears. The uploaded certificates are listed in the table.

Assign switches and install certificates

To install CA to bench router:

1. Navigate to **Admin > Certificate Management > CA Certificates**.
2. Locate the row with the uploaded CA certificate and click **Install** on a device, or click the ellipses (...) to lauch **Install** on a device.

   

   You can install unencrypted and encrypted keys and a certificate in a single bulk install; however, you must provide the key password used for encrypted keys.

3. For each certificate, click on the **Assign** arrow and select the switch to associate with the certificate.
4. Click **Install Certificates** to install all the certificates on their respective switches.

Unlink and delete certificates

After the certificates are installed on the switch, Nexus Dashboard cannot uninstall the certificate from Nexus Dashboard. However, you can always install a new certificate on the switch. The certificates that are not installed on the switches can be deleted. To delete the certificate installed on the switch, you must unlink the certificate from the switch, and then delete it from Nexus Dashboard.



Unlinking the certificate from the switch does not delete the certificate on the switch. The certificate still exists on the switch. Nexus Dashboard cannot delete the certificate on the Switch. To delete certificates from Nexus Dashboard repository, perform the following steps:

1. Select the certificate(s) that you need to delete.
2. From the **Actions** drop-down list, select **Unlink**.
3. Click **OK** to unlink the selected certificates from the switches.

   The switch name is removed from the **Attached To** column.

4. Select the certificate that is now unlinked from the Switch.
5. From the **Actions** drop-down list, select **Delete**.

   The certificate is deleted from Nexus Dashboard.

Bootstrap certificates

To provision switches using PnP or secure POAP method, ensure that you upload POAP/PnP server certificates on to Nexus Dashboard. This certificate is offered to Transport Layer Security (TLS) clients (switches).



Nexus Dashboard supports encrypted certificates only. Ensure that the POAP or PnP server certificate key is encrypted.

To upload or delete a bootstrap certificate:

1. Navigate to **Admin > Certificate Management > System Certificates**.
2. Choose a certificate from the **System Certificates** page
3. Click **Manage Feature Attachments**.
4. Click on the **Features** pull-down menu.
5. Choose **bootStrapServer**.
6. Click **Save**.

System certificates

Upload system certificates

To upload system certificates:

1. Navigate to the **Certificate Management** page:

   Admin > Certificate Management

2. Click the **System Certificates** tab, then click **Add system certificate** to upload the appropriate license file.
3. Browse your local directory and choose the certificate-key pair to upload.

   You can upload certificates with the .pem/.cer/.key/.crt/ file extensions.

4. Click **Save** to upload the chosen files to Nexus Dashboard.

   A successful upload message appears. The uploaded certificates are listed in the table.

> You must reload the browser URL once the new certificate is uploaded for webServer and Saved. It is possible that for a few seconds the Nexus Dashboard URL might show as disconnected.

Map a feature to a system certificate

After you have uploaded system certifications using the procedures provided in Upload system certificates, you can map a feature to a system certificate in the **System Certificates** page.

1. Navigate to the **Certificate Management** page:

   Admin > Certificate Management

2. Click the **System Certificates** tab.
3. Attach and install the certificate.

   - In the row **Manage Feature Attachments**, click the ellipse (...) and choose the feature to attach and install the certificate,

     or

   - Choose the certificate to see the **Manage Feature Attachments** button, then click to choose the feature to attach and install the certificate, then click **Save**.

   You can choose from these feature attachments:

   - WebServer
   - Bootstrap
   - Messagebug
   - Cyberark

Delete system certificates

You can delete system certificates after uploading new certificates.

1. Navigate to the **Certificate Management** page:

   Admin > Certificate Management

2. Click the **System Certificates** tab.
3. Click **Actions > Delete** to delete the certificate from Cisco Nexus Dashboard.

Fabric certificates

Enable NX-API certificate verification

The NX-API certificate verification is enabled using the toggle button on the **Fabric Certificates** page. However, this must be done only after all the switches managed by Cisco Nexus Dashboard are installed with CA-signed certificates and the corresponding CA Root certificates (one or more) are uploaded to Cisco Nexus Dashboard. When this is enabled, the Cisco Nexus Dashboard SSL client starts verifying the certificates that are offered by the switches. If the verification fails, the NX-API calls fail.



- Verification of the NX-API certificates cannot be enforced per switch; it is for either all or none. Hence, it is important that the verification is enabled only when all the switches have their corresponding CA-signed certificates installed.
- It is also required that all the CA certificates are installed on the Cisco Nexus Dashboard.
- When an NX-API call fails for a given switch because of verification issues, you can use the toggle button to disable enforcement, and all goes back to the previous state without any consequences.
- Because of the above mentioned points, you must enable the enforcement during a maintenance window.

To enable NX-API certificate verification:

1. Navigate to the **Certificate Management** window:

   Admin > Certificate Management

2. Click the **Fabric Certificates** tab, then click the toggle switch next to the **Enable NX-API certificate verification** field.

   A Warning popup appears, asking for verification that you want to perform this action.

3. Confirm that you have met all the guidelines described earlier in this section, then click **Enable** to enable NX-API certificate verification.

Fabric certificates

Guidelines and limitations: Fabric certificates

- When assigning fabric certificates to switches using one-to-one mapping, it may take a few minutes for the assignment task to complete.

Upload fabric certificates

To upload fabric certificates onto Nexus Dashboard, perform the following steps:

1. Navigate to the **Certificate Management** window:

Admin > Certificate Management

2. Click the **Fabric Certificates** tab, then click **Add fabric certificate** to upload the appropriate license file.
3. Browse your local directory and choose the certificate-key pair to upload.

   You can upload certificates with the .pem/.cer/.key/.crt/ file extensions.

4. Click **Save** to upload the selected files to Nexus Dashboard.

   A successful upload message appears. The uploaded certificates are listed in the table.

Delete fabric certificates

You can delete fabric certificates after uploading new certificates.

1. Navigate to the **Certificate Management** window:

   Admin > Certificate Management

2. Click the **Fabric Certificates** tab.
3. Click **Actions > Delete** to delete the certificate from Cisco Nexus Dashboard.